



HI-035

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of :  
Seung June YI and Jin Young PARK :  
Serial No.: 09/863,400 : Group Art Unit: To Be Assigned  
Filed: May 24, 2001 : Examiner: To Be Assigned  
For: METHOD FOR INSERTING LENGTH INDICATOR IN PROTOCOL  
DATA UNIT OF RADIO LINK CONTROL

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Assistant Commissioner of Patents  
Washington, D. C. 20231


Sir:

At the time the above application was filed, priority was claimed based on the  
following application:

Republic of Korea Application No. 48144/2000, filed August 19, 2000.

A copy of the priority application listed above is enclosed.

Respectfully submitted,  
FLESHNER & KIM, LLP

  
Daniel Y.J. Kim  
Registration No. 36,186  
Anthony H. Nourse  
Registration No. 46,121

P. O. Box 221200  
Chantilly, Virginia 20153-1200  
703 502-9440  
Date: July 13, 2001  
DYK:AHN/jad



대한민국 특허청  
KOREAN INTELLECTUAL  
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원번호 : 특허출원 2000년 제 48144 호  
Application Number

출원년월일 : 2000년 08월 19일  
Date of Application

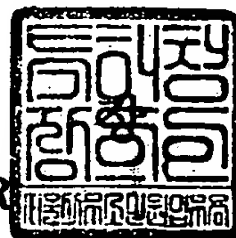
출원인 : 엘지정보통신주식회사  
Applicant(s)



2001 년 05 월 16 일

특 허 청

COMMISSIONER



CERTIFIED COPY OF  
PRIORITY DOCUMENT

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0002
【제출일자】	2000.08.19
【국제특허분류】	H04B
【발명의 명칭】	라디오 링크 콘트롤 ( R L C )에서 프로토콜 데이터 유닛( P D U ) 정보의 길이 인디케이터( L I ) 처리방법
【발명의 영문명칭】	METHOD FOR LENGTH INDICATOR INSERTING IN PROTOCOL DATA UNIT OF RADIO LINK CONTROL
【출원인】	
【명칭】	엘지정보통신 주식회사
【출원인코드】	1-1998-000286-1
【대리인】	
【성명】	허용록
【대리인코드】	9-1998-000616-9
【포괄위임등록번호】	1999-047173-5
【발명자】	
【성명의 국문표기】	이승준
【성명의 영문표기】	YI, Seung June
【주민등록번호】	720625-1025312
【우편번호】	135-240
【주소】	서울특별시 강남구 개포동 대청아파트 303동 403호
【국적】	KR
【발명자】	
【성명의 국문표기】	박진영
【성명의 영문표기】	PARK, Jin Young
【주민등록번호】	740628-2001315
【우편번호】	435-050
【주소】	경기도 군포시 금정동 무궁화 화성아파트 124동 1802호
【국적】	KR
【심사청구】	청구

## 【취지】

특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인  
허용록 (인)

## 【수수료】

【기본출원료】 20 면 29,000 원

【가산출원료】 1 면 1,000 원

【우선권주장료】 0 건 0 원

【심사청구료】 2 항 173,000 원

【합계】 203,000 원

## 【첨부서류】

1. 요약서·명세서(도면)\_1통

**【요약서】****【요약】**

본 발명은 RLC 계층의 PDU내에 SDU의 경계면이 있을 때 사용되는 LI의 정보 처리방법에 관한 것이다.

본 발명은 RLC에서 헤더와 LI그룹 및 데이터를 포함하는 PDU를 생성함에 있어 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입할 때, (a). 현재 PDU에 SDU의 마지막을 나타내 주는 LI값이 존재하고 그 때 해당 PDU의 끝과 SDU의 끝이 일치하는가, 또는 현재 PDU의 맨 마지막 부분이 남고 이 것을 패딩이라고 알려주기 위한 LI값을 삽입하였을 때 그 PDU의 끝과 SDU의 끝이 일치하는가를 검색하는 단계, (b). 상기 검색 결과 두가지 조건을 만족하지 않는 경우, 바로 다음 PDU의 LI그룹 선두에 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입하는 단계, (c). 상기 검색 결과 두가지 조건 중에서 어느 하나의 조건을 만족하는 경우, 바로 다음 PDU의 LI그룹 선두에는 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입하지 않는 단계; 로 이루어진다.

**【대표도】**

도 7

**【색인어】**

라디오 링크 프로토콜(RLC), 프로토콜 데이터 유닛(PDU), 길이 인디케이터

**【명세서】****【발명의 명칭】**

라디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU) 정보의 길이 인디케이터(LI) 처리방법{METHOD FOR LENGTH INDICATOR INSERTING IN PROTOCOL DATA UNIT OF RADIO LINK CONTROL}

**【도면의 간단한 설명】**

표1은 특정 목적을 위한 LI값을 7비트로 표현한 예를 나타낸 도표

표2는 특정 목적을 위한 LI값을 15비트로 표현한 예를 나타낸 도표

도1은 UMD PDU의 포맷을 나타낸 도면

도2는 AMD PDU의 포맷을 나타낸 도면

도3은 AMD PDU에서 7비트 LI를 사용하여 한 PDU내에서 SDU들의 끝부분을 나타내 주는 예를 나타낸 도면

도4는 AMD PDU에서 다음 PDU에 LI='0' 값이 필요한 경우를 설명하기 위한 도면

도5는 종래의 LI 처리방법을 따를 때 불필요한 LI를 삽입하게 되는 경우를 설명하기 위한 도면

도6은 종래의 LI 처리방법을 따를 때 불필요한 LI를 삽입하게 되는 경우의 다른 예를 설명하기 위한 도면

도7은 본 발명의 라디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법을 나타낸 플로우차트

## 【발명의 상세한 설명】

## 【발명의 목적】

## 【발명이 속하는 기술분야 및 그 분야의 종래기술】

<10> 본 발명은 RLC 계층의 PDU내에 SDU의 경계면이 있을 때 사용되는 LI의 정보 처리방법에 관한 것으로서, 특히 PDU에 삽입되는 LI가 불필요하게 포함되지 않도록 LI값의 삽입 여부를 선택적으로 제어하는 라디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법에 관한 것이다.

<11> 더욱 상세하게는 본 발명은 RLC에서 PDU를 생성함에 있어서 LI값을 삽입할 때, 이전 PDU의 SDU의 마지막을 나타내 주는 LI가 있음에도 불구하고 불필요하게 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 맞았음을 나타내 주는 LI값(LI='0')을 넣어주는 경우나, PDU의 맨 마지막 부분이 한 옥텟 남을 경우 이를 패딩이라고 알려주기 위한 LI값(LI='1')을 넣어준 경우에도 불필요하게 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 맞았음을 나타내 주는 LI값(LI='0')을 넣어주는 경우를 방지할 수 있도록 한 라디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법에 관한 것이다.

<12> 이른 바 멀티미디어의 시공간적 제약없는 접근을 허용하는 통신기술의 연구와 그 연구의 가시적 성과를 바라는 많은 노력들이 경주되고 있는 현실에 비추어 볼 때, 디지털 데이터 처리와 전송 기술의 발달은 유선과 무선 통신을 통합하고 인공위성을 이용한 실시간 글로벌 데이터 통신 시스템의 구현을 눈앞에 두고 있다.

<13> 또한 이와같은 디지털 데이터의 처리와 전송 기술의 발달에 힘입어 기존의 음성 통

화는 물론 네트워크 기반의 정지화상, 동화상의 실시간 전송과 유무선을 가리지 않고 언제 어느 곳에서나 자유로운 정보의 접근을 가능하게 하고 있다.

<14> IMT-2000은 그 중의 하나가 될 것이다.

<15> 본 발명에서 언급되는 RLC(Radio Link Control) 계층은 3GPP의 제2계층으로서, 수신측에서 PDU(Protocol Data Unit)를 받은 후 송신측으로의 인식 신호가 필요없는 경우에 사용되는 UMD PDU(Unacknowledged PDU)와 인식 신호가 필요한 경우에 사용되는 AMD PDU(Acknowledged PDU)의 두가지 형태의 PDU가 존재하며, 각각의 PDU 포맷은 도1 및 도2에 나타낸 바와같다.

<16> UMD PDU의 포맷은 도1에 도시한 바와같이 헤더(header)와 LI 부분(Length Indicator group), 데이터(Data), PAD(padding)로 이루어진다. 헤더는 각 PDU의 순서번호를 나타내는 필드인 12비트의 시퀀스 넘버(Sequence Number)와, 그 다음 필드가 데이터 인지 아니면 LI와 E 비트인지를 알려주기 위한 E(Extension) 필드 1비트로 이루어진다. 데이터 부분은 상위 계층에서 내려온 SDU들에 해당하는 필드로서 하나 또는 여러개의 SDU를 포함하여, 이러한 데이터 부분은 그 크기가 가변적이기 때문에 전체 PDU 사이즈를 옥텟-정렬(octet align)하기 위해서 패딩(padding)을 한다.

<17> AMD PDU의 포맷은 도2에 도시한 바와같이 헤더와 LI부분, 데이터를 포함하며, 상기 UMD PDU에서의 패딩 대신에 피기백(piggyback) 타입의 상태 PDU(STATUS PDU)를 삽입하여 전송이 가능하도록 하고 있다.

<18> AMD PDU에서 헤더에는 시퀀스 넘버 이외에, 해당 PDU가 데이터 정보를 싣고 있는지 혹은 콘트롤 정보를 싣고 있는지를 알려주는 1비트의 D/C필드와, 수신측에 상태 리포트



(status report)를 요구하는 1비트의 폴링(Polling) 필드 - P필드와, 다음의 데이터가 데이터인지 아니면 LI와 E 비트인지를 알려주는 2비트의 HE(Header Extension) 필드가 추가되어 있다.

- <19>       상기한 UMD PDU와 AMD PDU에서 LI 부분은 LI와 E 비트로 구성되는데, LI는 그 PDU가 여러개의 SDU를 포함할 경우 각 SDU의 경계면을 나타내는 필드이다. 각 LI는 데이터 부분의 첫 옥텟부터 각 SDU의 끝 옥텟까지의 옥텟 수를 나타내며, 한 PDU에 포함된 SDU들에 대한 각각의 LI들을 LI그룹이라고 한다.
- <20>       LI의 사이즈는 7비트 또는 15비트가 사용된다. AMD PDU의 경우에는 PDU의 사이즈가 126옥텟보다 같거나 작은 경우에는 7비트 LI를 사용하고, 그렇지 않은 경우에는 15비트 LI를 사용한다. UMD PDU의 경우에는 PDU 사이즈가 125 옥텟 보다 같거나 작은 경우에는 7비트 LI를 사용하고, 그렇지 않은 경우에는 15비트 LI를 사용한다.
- <21>       이러한 LI들은 특정한 목적에 사용하기 위해서 몇개의 값들은 미리 정의되어 있다. 표1은 특정한 목적을 위한 LI값(7비트 LI인 경우)을 예시하고 있으며, 표2는 15비트 LI인 경우에 특정한 목적을 위한 LI값을 보이고 있다.
- <22>       표1 및 표2에서 보는 바와같이, 만약 한 SDU의 끝 부분이 그 PDU의 마지막 부분에 정확히 맞는 경우에는 이 것을 나타내 주기 위해서 바로 다음 PDU의 첫번째 LI의 값을 'LI=0'으로 삽입한다. 그리고 LI가 15비트 길이인 경우에 있어서 RLC SDU의 마지막 세그먼트가 PDU의 끝에서 한 옥텟 모자라는 경우에는 이 것을 나타내 주기 위해서 바로 다음 PDU의 첫번째 LI에 '111 1111 1111 1011' 값이 들어간다. 또한 사용되지 않는 공간인 패딩(padding)을 포함하는 PDU는 RLC PDU의 나머지 부분이 패딩임을 나타내 주기 위한 값으로 'LI=1'을 사용하며, 이 때 패딩은 반드시 RLC PDU의 맨 뒷부분에 위치한다.

## &lt;23&gt; 【표 1】

Bit	설 명
0000000	바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 맞음
1111100	Reserved
1111101	Reserved
1111110	AM PDU: PDU의 나머지 부분에 piggybacked STATUS PDU가 포함됨 UM PDU: Reserved
1111111	PDU의 나머지 부분이 패딩임

## &lt;24&gt; 【표 2】

Bit	설 명
0000000000000000	바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 맞음
111111111111011	바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝에서 한 octet 모자람
111111111111100	Reserved
111111111111101	Reserved
111111111111110	AM PDU: PDU의 나머지 부분에 piggybacked STATUS PDU가 포함됨 UM PDU: Reserved
111111111111111	PDU의 나머지 부분이 패딩임

<25> 도3은 AMD PDU에서 7비트 LI를 사용해서 한 PDU내에서 SDU들의 끝 부분을 나타내 주는 예를 보이고 있다. 여기서 PDU 사이즈는 35옥텟이고 이 PDU에 포함된 데이터 사이즈는 24옥텟인 경우이다.

<26> 도3에서 보는 바와같이 이 AMD PDU는 3개의 SDU(SDU1, SDU2, SDU3)를 가지고 있고, 각각의 SDU들의 사이즈는 각각 11옥텟, 9옥텟, 4옥텟이다. SDU1, SDU2, SDU3 각각의 데이터 부분의 첫 옥텟부터 끝 옥텟까지의 옥텟 수를 지시하는 LI값은 각각 11(옥텟), 20(옥텟), 24(옥텟)로 삽입되었고, 'LI=111 1111'이 더 삽입되어 PDU의 나머지 부분이 패딩(padding)(5옥텟)이라는 것을 표현해 주고 있다.

<27> 그런데, 종래에는 RLC PDU가 한 SDU의 끝 부분이 그 PDU의 끝 부분에 정확하게 맞

는 모든 경우에 대하여 바로 다음 PDU에 LI='0'이 들어가게 된다.

<28> 도4, 도5, 도6은 종래의 LI처리 방법에 따라 LI='0'이 삽입되는 세가지 경우를 보이고 있는데, 도4의 경우는 올바른 처리이지만, 도5 및 도6과 같은 경우는 불필요한 LI값 삽입의 문제점을 보이고 있다.

<29> 먼저, 도4의 예는 다음 PDU에 LI='0'을 붙여주는 것이 의미를 가지게 된다. 이것은 도4에서 SDU2를 위한 LI가 현재의 PDU에 포함되게 되면 그 LI 때문에 다음 PDU로 SDU2의 한 옥텟 데이터가 넘어가야 하는 상황이 발생하기 때문이다. 그렇게 되면 실제로 SDU2의 끝이 해당 PDU의 끝과 맞지 않게 되므로 그러한 LI를 붙여주는 의미가 없게 된다. 따라서 이러한 경우에는 해당 PDU에 SDU2의 끝을 알려주는 LI를 넣는 대신 다음 PDU의 맨 처음 LI에 LI='0'을 넣어주는 것이 의미가 있게 된다.

<30> 그러나, 도5, 도6과 같은 경우에는 해당 PDU만으로도 다음 PDU로 이어지는 SDU가 포함되지 않았다는 점을 알 수 있기 때문에 다음 PDU에 LI='0'을 붙여 주지 않아도 된다.

<31> 즉, 도5에서 설명하고 있는 것은, 이전 PDU의 SDU2의 마지막을 나타내 주는 LI가 있는데도 불구하고 불필요하게 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치하였음을 나타내주는 LI='0'값을 바로 다음 PDU에 넣어주고 있다는 의미이다.

<32> 또한 도6에서 설명하고 있는 것은, 해당 PDU(도6의 그림에서 가운데 PDU)의 맨 마지막 부분이 한 옥텟 남기 때문에 이 것을 패딩이라고 알려주기 위한 LI값 즉, LI='1'값을 넣어주는데(그러나 이 것을 넣어 줌으로써 패딩의 자리는 없어지므로 '0' 패딩이 됨), 이미 LI='1'값에 의해서 해당 PDU의 끝과 SDU2의 마지막 세그먼트가 정확히 일치하

였음을 의미하게 되므로, 바로 다음의 PDU에서 LI='0'값을 넣어주는 것은 불필요한 것이다.

- <33> 앞에서 설명한 문제점은 AMD PDU 뿐만 아니라 UMD PDU에서도 동일하게 나타나며, 따라서 종래의 LI삽입 처리 방법에 따르면 불필요하게 LI값(LI='0')을 PDU의 LI그룹에 삽입하는 경우가 발생하게 되고, 이 것은 곧 불필요하게 중복되는 정보를 포함시켜 망 자원을 낭비시키는 결과를 초래하게 된다.

**【발명이 이루고자 하는 기술적 과제】**

- <34> 본 발명은 RLC에서 PDU를 생성할 때 바로 전 PDU내의 SDU의 끝과 바로 전 PDU의 끝 부분이 정확히 일치한 경우에도 해당 내용을 나타내 주기 위한 LI값의 삽입 여부를 선택적으로 적용함으로써, 불필요하게 LI값이 삽입되는 것을 방지할 수 있도록 한 래디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법을 제안한다.

- <35> 본 발명은 RLC에서 PDU를 생성함에 있어서, 바로 전 PDU내의 SDU의 끝과 바로 전 PDU의 끝부분이 정확히 일치하였을 때, 바로 전 PDU에 바로 전 SDU의 끝을 나타내는 LI를 넣어줄 수 없는 경우에는 상기 LI값을 다음 PDU의 LI그룹 선두에 넣어주고, 이전 PDU에 바로 전 SDU의 마지막을 나타내 주는 LI가 있는 경우에는 바로 다음 PDU의 LI그룹에 상기 LI값(LI='0')을 넣어주지 않고, 해당 PDU의 맨 마지막 부분이 한 옥텟 남아서 이를 패딩이라고 알려주기 위한 LI값(LI='1')을 넣어준 경우에는 바로 다음 PDU의 LI그룹에 상기 LI값(LI='0')을 넣어주지 않는 방법으로 LI값을 처리함으로써, 불필요하게 LI값이 삽입되는 것을 방지할 수 있도록 한 래디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법을 제안한다.

## 【발명의 구성 및 작용】

<36> 본 발명은 RLC에서 헤더와 LI그룹 및 데이터를 포함하는 PDU를 생성함에 있어 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입할 때, (a). 현재 PDU에 SDU의 마지막을 나타내 주는 LI값이 존재하고 그 때 해당 PDU의 끝과 SDU의 끝이 일치하는가, 또는 현재 PDU의 맨 마지막 부분이 남고 이 것을 패딩이라고 알려주기 위한 LI값을 삽입하였을 때 그 PDU의 끝과 SDU의 끝이 일치하는가를 검색하는 단계, (b). 상기 검색 결과 두가지 조건을 만족하지 않는 경우, 바로 다음 PDU의 LI그룹 선두에 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입하는 단계, (c). 상기 검색 결과 두가지 조건 중에서 어느 하나의 조건을 만족하는 경우, 바로 다음 PDU의 LI그룹 선두에는 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입하지 않는 단계; 로 이루어진 것을 특징으로 하는 래디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법 이다.

<37> 또한 본 발명은 RLC에서 헤더와 LI그룹 및 데이터를 포함하는 PDU를 생성함에 있어 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입할 때,

<38> (a). 현재 PDU에 SDU<sub>i</sub>의 끝이 포함됨과 함께, SDU<sub>i</sub>의 끝 위치에 대한 LI를 넣은 후에는 SDU<sub>i</sub>가 해당 PDU에 포함될 수 없는 경우에는 바로 다음 PDU의 첫번째 LI에 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입하는 단계,

<39> (b). 현재 PDU에 SDU<sub>i</sub>의 끝이 포함됨과 함께 SDU<sub>i</sub>의 끝 위치에 대한 LI를 넣은 후

에도 SDUi가 해당 PDU에 포함될 수 있는 경우에는 SDUi의 끝 위치에 대한 LI값을 현재 PDU에 삽입하는 단계,

<40> (c). 상기 SDUi의 끝 위치에 대한 LI값을 현재 PDU에 삽입한 후의 PDU 사이즈가 현재까지의 데이터와 LI그룹 및 헤더 사이즈의 합과 다르고 현재 전송할 SDU가 존재하지 않을 경우, 해당 PDU에 PDU의 나머지 부분이 패딩임을 의미하는 LI값을 넣고 해당 PDU의 나머지 부분을 패딩 처리하는 단계; 로 이루어진 것을 특징으로 하는 래디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법 이다.

<41> 상기한 바와같이 이루어진 본 발명의 래디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법에서는, 도5 및 도6과 같은 경우에 해당할 때에는 다음 PDU의 LI그룹 선두에 LI='0'을 넣지 않는 것을 특징으로 한다. 즉, 본 발명에서는 도4와 같은 경우에만 다음 PDU의 LI그룹 선두에 LI='0'을 넣고, 도5나 도6과 같은 경우에는 다음 PDU의 LI그룹 선두에 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값(LI='0')을 넣는다.

<42> 따라서 본 발명에 의하면 SDU의 끝이 PDU의 끝과 맞는 여러가지 경우를 모두 고려하여 다음 PDU에 LI='0'을 넣지 않아도 해당 PDU에서 이미 SDU의 끝을 알 수 있는 경우에 대해 불필요하게 중복되는 LI가 포함되는 것을 방지할 수 있게 된다.

<43> 도7은 상기한 본 발명의 래디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법의 플로우차트 이다.

<44> 도7을 참조하여 본 발명을 더욱 상세하게 설명한다.

<45> 먼저, 첫번째 단계(701)는 현재 PDU에 SDUi의 끝이 포함될 수 있는가를 판단한다.

이 단계(701)에서 현재 PDU에 SDUi의 끝이 포함될 수 없다면 다음 단계(702)로 이행하여 PDU의 크기에 맞게 SDUi의 세그먼트를 넣고 종료한다.

<46>      상기 단계(701)에서 현재 PDU에 SDUi의 끝이 포함될 수 있다면 다음 단계(703)로 이행하여 SDUi의 끝 위치에 대한 LI값을 넣은 후에도 SDUi가 해당 PDU에 포함될 수 있는가를 판단한다.

<47>      상기 단계(703)의 판단 결과 SDUi의 끝 위치에 대한 LI값을 넣은 후에는 SDUi가 해당 PDU에 포함될 수 없다면 다음 단계(704)로 이행하여 바로 다음 PDU의 첫번째 LI에 LI='0'(바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 정보)을 넣고 종료한다(즉, 이 경우는 도4와 같은 경우가 될 것이다).

<48>      상기 단계(703)의 판단 결과 SDUi의 끝 위치에 대한 LI값을 넣은 후에도 SDUi가 해당 PDU에 포함될 수 있다면 그 다음 단계(705)로 이행하여 SDUi의 끝 위치에 대한 LI값을 넣고 PDU 사이즈가 현재까지의 데이터와 LI그룹 및 헤더 사이즈의 합과 동일한가를 판단한다(단계 706).

<49>      상기 단계(706)의 판단 결과 PDU 사이즈가 현재까지의 데이터와 LI그룹 및 헤더 사이즈의 합과 같다면 다음 PDU에 LI='0'을 넣을 필요가 없는 경우 즉, 도5와 같은 경우이므로 종료한다.

<50>      상기 단계(706)의 판단 결과 PDU 사이즈가 현재까지의 데이터와 LI그룹 및 헤더 사이즈의 합과 같지 않은 경우에는 현재 보낼 SDU가 더 존재하는가를 판단하고(단계 707), 현재 보낼 SDU가 더 존재하는 경우에는 다음 단계(708)로 이행하여 i값을 +1 증가시켜 다음 순번의 SDU에 대해서 상기 첫번째 단계(701)부터 반복 수행한다.

<51> 그러나, 현재 보낼 SDU가 더 이상 존재하지 않는 경우에는 다음 단계(709)로 이행하여, 해당 PDU에 LI='1'값(즉, PDU의 나머지 부분이 패딩임을 의미하는 값)을 넣고 해당 PDU의 나머지 부분에 패딩을 넣는 것으로 종료한다. 즉, 이 경우는 도6과 같은 경우가 되며, 그 다음 PDU의 LI그룹 선두에 값이 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 정보인 LI='0'값을 넣지 않아도 되는 것이다.

**【발명의 효과】**

<52> 본 발명은 종래의 RLC PDU에서 한 SDU의 끝이 PDU의 끝과 맞는 모든 경우에 대해 바로 다음에 오는 PDU의 LI그룹에 LI='0' 이 붙는 LI 처리방법에 비하여, 해당 PDU에서 이미 SDU의 끝을 알 수 있는 경우를 찾고, 그러한 경우에 해당하면 해당 PDU 바로 다음에 오는 PDU의 LI그룹에 LI='0'을 넣지 않는 처리방법을 제안함으로써, 불필요하게 LI값이 삽입되는 것을 방지할 수 있다.

<53> 또한 본 발명에 따르면 불필요한 LI값 삽입이 배제되므로 망 자원의 낭비를 막을 수 있을 뿐만 아니라, 불필요한 LI값 까지 처리하는데 드는 오버헤드도 줄일 수 있는 효과가 있다.



**【특허청구범위】****【청구항 1】**

RLC에서 헤더와 LI그룹 및 데이터를 포함하는 PDU를 생성함에 있어 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입할 때, (a). 현재 PDU에 SDU의 마지막을 나타내 주는 LI값이 존재하고 그 때 해당 PDU의 끝과 SDU의 끝이 일치하는가, 또는 현재 PDU의 맨 마지막 부분이 남고 이 것을 패딩이라고 알려주기 위한 LI값을 삽입하였을 때 그 PDU의 끝과 SDU의 끝이 일치하는가를 검색하는 단계, (b). 상기 검색 결과 두가지 조건을 만족하지 않는 경우, 바로 다음 PDU의 LI그룹 선두에 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입하는 단계, (c). 상기 검색 결과 두가지 조건 중에서 어느 하나의 조건을 만족하는 경우, 바로 다음 PDU의 LI그룹 선두에는 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입하지 않는 단계; 로 이루어진 것을 특징으로 하는 래디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법.

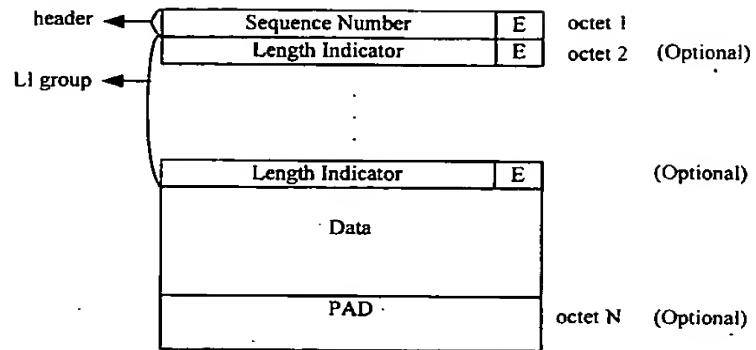
**【청구항 2】**

RLC에서 헤더와 LI그룹 및 데이터를 포함하는 PDU를 생성함에 있어 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입할 때, (a). 현재 PDU에 SDU의 끝이 포함됨과 함께, SDU의 끝 위치에 대한 LI를 넣은 후에는 SDU가 해당 PDU에 포함될 수 없는 경우에는 바로 다음 PDU의 첫번째 LI에 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 일치함을 나타내는 LI값을 삽입하는 단계, (b). 현재 PDU에 SDU의 끝이 포함됨과 함께 SDU의 끝 위치에 대한 LI를 넣은 후

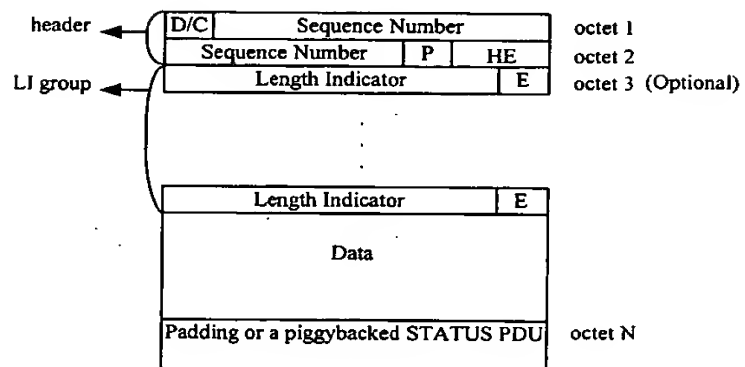
에도 SDUi가 해당 PDU에 포함될 수 있는 경우에는 SDUi의 끝 위치에 대한 LI값을 현재 PDU에 삽입하는 단계, (c). 상기 SDUi의 끝 위치에 대한 LI값을 현재 PDU에 삽입한 후의 PDU 사이즈가 현재 까지의 데이터와 LI그룹 및 헤더 사이즈의 합과 다르고 현재 전송할 SDU가 존재하지 않을 경우, 해당 PDU에 PDU의 나머지 부분이 패딩임을 의미하는 LI값을 넣고 해당 PDU의 나머지 부분을 패딩 처리하는 단계; 로 이루어진 것을 특징으로 하는 래디오 링크 컨트롤(RLC)에서 프로토콜 데이터 유닛(PDU)의 길이 인디케이터(LI) 처리방법.

## 【도면】

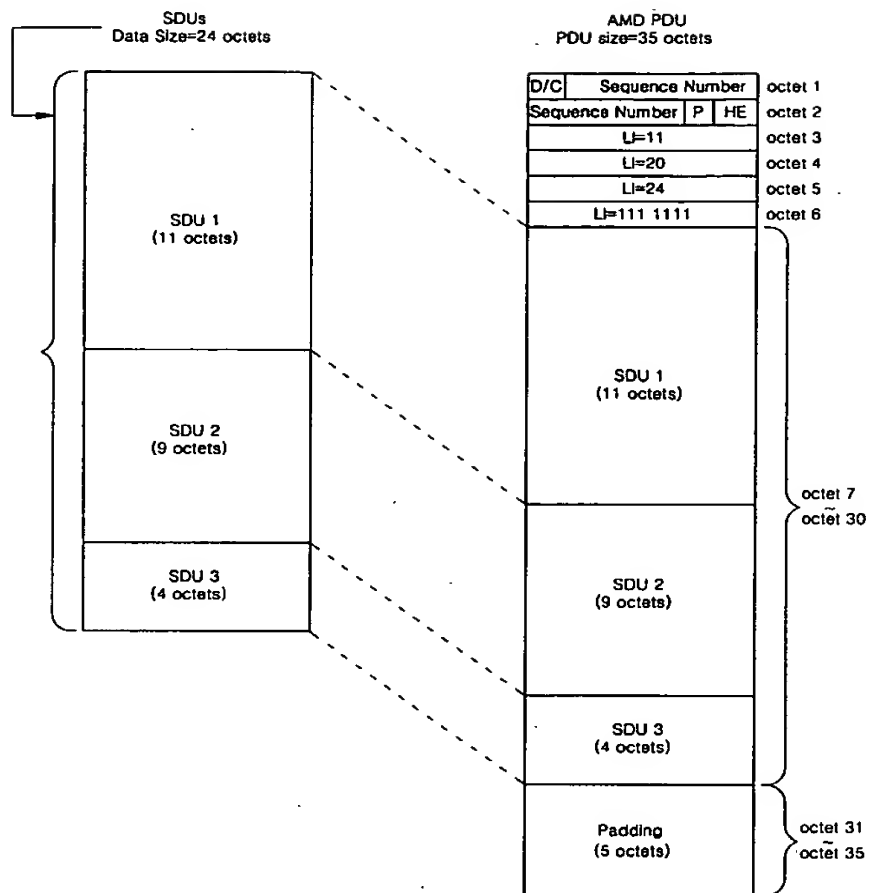
【도 1】



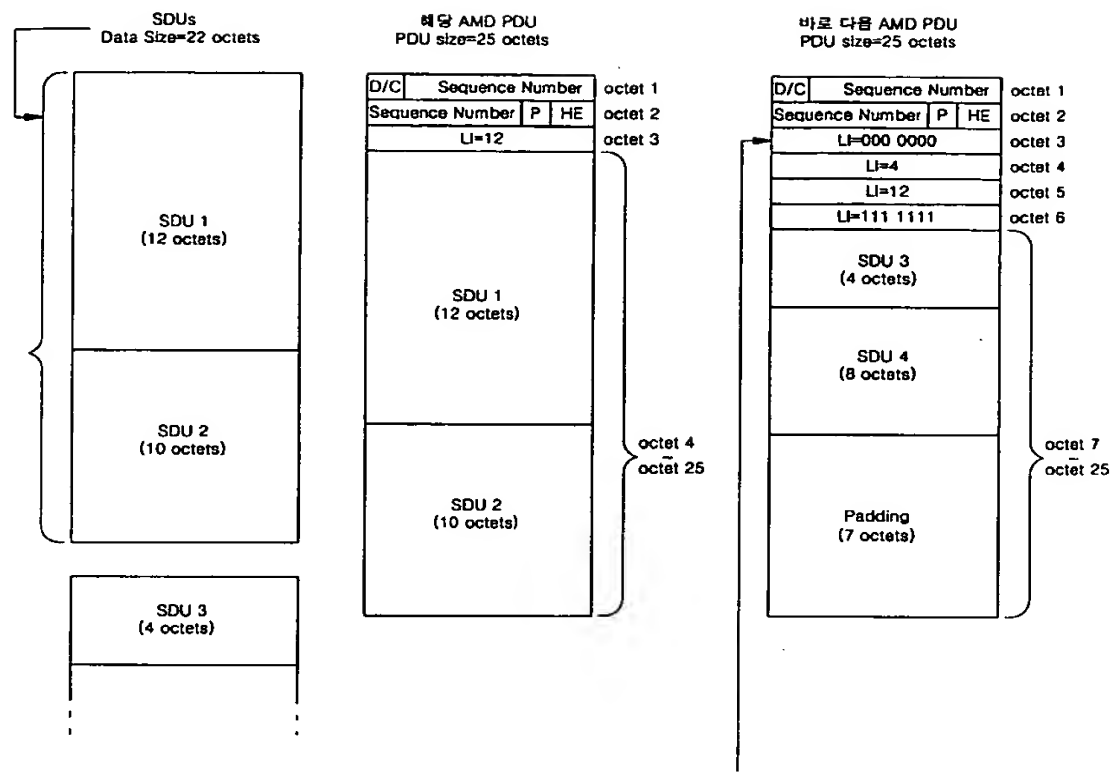
【도 2】



【도 3】

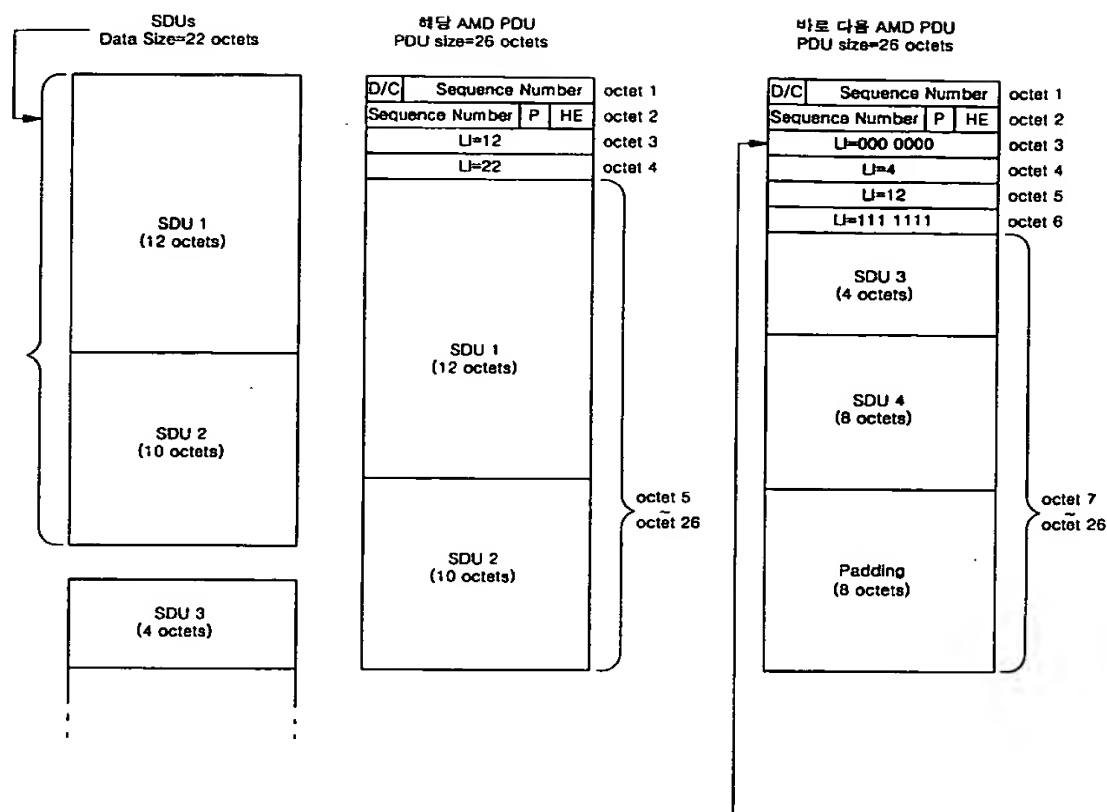


【도 4】



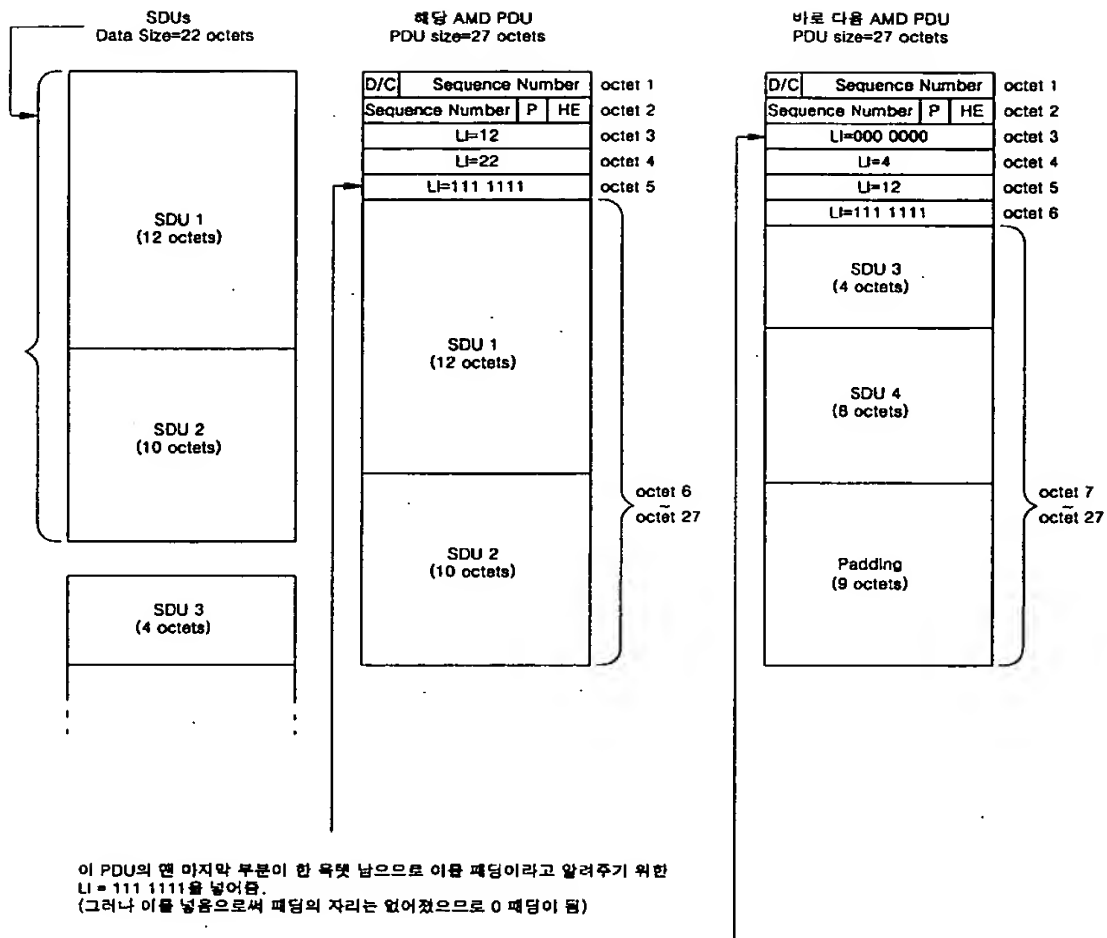
바로 전 PDU에 SDU2의 끝을 나타내는 LI를 넣어줄 수 없으므로 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 맞았음을 나타내 주는 LI = 000 0000 을 넣어주게 됨

【도 5】



이전 PDU에 SDU2의 마지막을 나타내 주는 LI가 있는데도 불구하고 불필요하게  
바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 맞았음을 나타내  
주는 LI = 000 0000 을 넣어주게 됨

【도 6】



불필요하게 바로 전 SDU의 마지막 세그먼트가 바로 전 PDU의 끝과 정확히 맞았음을 나타내 주는 LI = 000 0000 을 넣어주게 됨

【도 7】

